Inventor(s): WILLIAM V. DA PALMA
              BRETT J. GAVAGNI
              MATTHEW W. HARTLEY
              BRIEN H. MUSCHETT

METHOD AND SYSTEM FOR DYNAMIC CONDITIONAL INTERACTION IN A VOICEXML RUN-TIME SIMULATION ENVIRONMENT

BACKGROUND OF THE INVENTION

Statement of the Technical Field

[0001]   The present invention relates to the field of computer speech recognition, text-to-speech technology and telephony, and more particularly to a system and method for a run-time simulation environment for voice applications that simulates and automates user interaction.

Description of the Related Art

[0002]   Functionally testing voice applications presents many difficulties. In the case of a VoiceXML (VXML) application, a VXML interpreter communicates with a platform that supplies the necessary speech technology needed to test the application in real-time. These speech technologies, such as an automatic speech recognition (ASR) engine, or a text-to-speech (TTS) engine or converter, are generally very CPU intensive and expensive to build and install. In addition to the speech technologies, to test a

15071   BOC9-2003-0115

voice application a tester must also provided the input to the application. This usually requires a tester to physically perform the interaction, in the form of actual speech or key tone input, which may be cumbersome and difficult to provide. Having a person perform the input can be time consuming and costly.

[0003] Furthermore, when testing a voice application, it can be difficult to mimic the true behavior of speech or audio input to the application, as well as any text-to-speech or pre-recorded audio output from the application. When testing voice applications, it may be necessary to test for dynamic and conditional interaction between the voice application dialog and the user. For example, the voice application dialog may prompt a user for an input, which input may vary according to certain conditions existing at the time the user makes the input.

[0004] It would be desirable therefore to provide a testing environment that allows the simulation of user interaction as well as the simulation of the speech technology platform, such that a developer of voice applications will no longer be dependent on human testers and speech technology and hardware to test their applications. The testing environment would therefore be a "simulation environment" that would adequately replace the user and speech technologies. To simulate a robust voice application, it would be necessary to provide a simulation environment that allowed for user interaction under varying conditions. It would be desirable therefore, to provide a simulation environment that could simulate conditional user interaction.

SUMMARY OF THE INVENTION

[0005] The present invention addresses the deficiencies of the art in respect to testing voice applications and provides a novel and non-obvious method, system and apparatus for a dynamic run-time simulation environment for voice applications that simulates and automates conditional user interaction.

[0006] Methods consistent with the present invention provide a method for simulating a dynamic run-time user interaction with a voice application. A user simulation script programmed to specify simulated voice interactions with the voice application is loaded. The voice application is first processed to derive a nominal output of the voice application. The user simulation script is second processed to generate a simulated output for the voice application corresponding to the nominal output. Next, the user simulation script is third processed to generate a first simulated input for the voice application corresponding to a first pre-determined user input to the voice application, if the nominal output satisfies a first condition. Or, the user simulation script is fourth processed to generate a second simulated input for the voice application corresponding to a second pre-determined user input to the voice application, if the nominal output satisfies a second condition different from the first condition.

[0007] Systems consistent with the present invention include a simulation tool for simulating a dynamic run-time user interaction with a voice application running on an application server. The tool is configured to load a user simulation script programmed to specify simulated voice interactions with the voice application and to process the voice application to derive a nominal output of the voice application. The tool is further

15071   BOC9-2003-0115

3

configured to process the user simulation script to generate a simulated output for the voice application corresponding to the nominal output. If the nominal output satisfies a first condition, the tool is configured to process the user simulation script to generate a first simulated input for the voice application corresponding to a first pre-determined user input to the voice application. If the nominal output satisfies a second condition different from the first condition, the tool is configured to generate a second simulated input for the voice application corresponding to a second pre-determined user input to the voice application,.

[0008] Additional aspects of the invention will be set forth in part in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The aspects of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]   The accompanying drawings, which are incorporated in and constitute part of the this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention. The embodiments illustrated herein are presently preferred, it being understood, however, that the invention is not limited to the precise arrangements and instrumentalities shown, wherein:

[0010]   Figure 1 is a conceptual drawing of the present invention which provides a user interaction simulation environment for a voice application;

[0011]   Figure 2 is a block diagram showing the arrangement of elements in a system assembled in accordance with the principles of the present invention for simulating a run-time environment with a voice application;

[0012]   Figure 3 is a flowchart illustrating a process for simulating a run-time user interaction with a voice application; and

[0013]   Figure 4 is a flowchart illustrating a process for simulating conditional user interaction with a voice application.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0014] The present invention is a system and method for simulating a run-time user interaction with a voice application. Figure 1 is a conceptual drawing of the present invention which provides a user interaction simulation environment for a voice application. The simulation environment 100 of the present invention includes a simulation tool 101 that is coupled to a voice application 105. The simulation tool 101 uses conditional logic to process conditional statements in a simulation script 110 that provides a set of specified inputs and outputs to and from the voice application, to simulate a real-time interaction by a user with the voice application. The simulation tool 101 and script 110 replace the actual inputs that may be provided by a live user, and replace the actual outputs that may be provided by the voice application 101 and all the speech technologies that are otherwise coupled to a conventional voice application.

[0015] As used herein, a "voice application" shall mean any logic permitting user interaction through a voice driven user interface, such as a mark-up language specification for voice interaction with some form of coupled computing logic. One example of a voice application is an application written in Voice Extensible Mark-up Language, or "VoiceXML." However, it is readily understood that VoiceXML applications are not the only type of voice applications, and any reference to the term "VoiceXML application" herein shall encompass all voice applications.

[0016] In conventional voice systems, the voice application itself receives the "outputs" it generates to users from various speech technologies coupled to the voice application. For example, the voice application can receive an input from the user, and

can record the input with an audio device, or convert the spoken word input into text using an automatic speech recognition engine. The voice application can then playback the recorded audio to the user as a prompt, or may convert a text stream to audio using the text-to-speech capabilities of a speech technologies platform, either of which may be sent as another "output" to the user.

[0017] Heretofore, to test a voice application, all of the foregoing speech processing elements are needed. The present invention replaces a number of those elements, by providing a simulation environment that allows a voice application to be executed in real-time, and that supplies and simulates the execution time of the inputs and outputs that flow to and from the voice application. Furthermore, the simulated inputs provided by the simulation environment can utilize conditional statements and conditional logic to provide a dynamic interaction with the voice application.

[0018] The present invention is a method, system and apparatus for dynamic conditional interaction in a voice application run-time simulation environment. In accordance with the present invention, a user simulation script for exercising the run time environment of a voice application can be provided. The simulation script can be processed by a simulation script interpreter to provided simulated audible input into the voice application in order to test the operation of the voice application without requiring a human applications tester to manually speak input into the voice application. Importantly, the simulation script can include one or more conditional statements that can be resolved by applying conditional logic. In this regard, the conditional tags can trigger a conditional statement in the script interpreter in which the input provided to the voice application can vary based upon the resolution of the conditional logic.

[0019] In further illustration of the inventive arrangements, Figure 2 is a block diagram illustrating a system for dynamic conditional interaction in a voice application run-time simulation environment. The system 200 can include a voice application interpreter 202 operating in association with a voice application 201. The voice application interpreter 202 can be configured to process the voice application 201 comprised of instructions for directing the management of voice interactions with an end user and application logic disposed within an application server (not shown). The system 200 also includes a simulation script 205 that can be interpreted by a second interpreter 210. The second interpreter 210 may reside on a separate piece of hardware, or may be resident on the same hardware as the voice application 201 and interpreter 202.

[0020] The simulation environment 200 can process customized mark-up language documents which describe the user interaction or the user experience with the environment itself. Specifically, the mark-up language documents describe the set of operations a user might take as a transcript of what occurs when interacting with the voice application. In this regard, what is the desired to be simulated is the behavior between the user and the voice application, which is provided by the simulation script 205 written in the customized mark-up language, which, by way of non-limiting example, may be called a "Voice User Interaction Extensible Mark-up Language," or "VuiXML." The user behavior, as well as the prompts and outputs supplied from the voice application itself, is mimicked and embodied in the user simulation script 205.

[0021] The user simulation script 205 can be a script that describes how the user interacts with the system. Common interaction behaviors can include voice response, input in the form of digits, pauses between spoken words, hang-up operations, typical

inputs that a user would make when interacting with a voice response system. This user interaction is embodied in the script.

[0022]   Figure 3 is a flowchart illustrating a process for simulating a run-time user interaction with a voice application. First, the voice application browser, such a VoiceXML browser, is called in step 301. Next, in step 305, a user simulation script is provided and supplied to the simulation environment. Subsequently, the voice application is processed in step 310.

[0023]   The voice application normally generates one or more outputs, which, in conventional systems, may be prompts, synthesized text to speech, pre-recorded audio, and the like. However, in the simulation environment, all such outputs are text based, and are initially "nominal" outputs: the outputs that the voice application would otherwise provide to a user in the non-simulated environment. Within the simulation environment, the actual outputs for the voice application are instead generated by the user simulation script, which generates a simulated output for the voice application corresponding to the nominal output. This occurs in step 315.

[0024]   In step 320, the process next determines whether the voice application requires a user input. Should the voice application require a user input, the user simulation script is processed in step 325 to generate a simulated input for the voice application corresponding to a pre-determined user input to the voice application. As stated above, all such input is pre-developed and supplied in the user simulation script. The process may then choose to continue after assessing whether additional

processing of the voice application is necessary in step 330, or may terminate if execution of the voice application is complete.

[0025] In accordance with the present invention, the voice application simulation script interpreter can be further configured to process conditional operations within the voice application simulation script. In this regard, the voice application simulation script interpreter can be configured to process one or more conditional tags disposed within the voice application simulation script such as "<if>", "<else>", and "<elseif>". When encountering such conditional tags, the voice application simulation script interpreter can invoke conditional logic to resolve a suitable interaction to be performed in respect to the voice application.

[0026] Referring back to FIG. 1, the simulation tool 101 of the present invention creates a dynamic run-time environment for user interaction with a voice application by providing a user simulation script 110 that includes conditional tags and an internal variable for the output from the voice application. The user simulation script therefore includes one or more conditional statements that are resolved by applying conditional logic. One or more conditional tags, such as "<if>", "<else>", or "<elseif>" can be used to trigger a conditional statement containing one or more logical tests, which when resolved, produces a varying result that depends on the outcome of the logical test in the conditional statement.

[0027] To further illustrate the conditional logic of the voice application, Figure 4 shows a flow chart illustrating a process for dynamically conditioning interactions in the voice application in the run-time simulation environment of Figure 1. First, the voice

application browser, such as the VoiceXML browser, is called in step 401. In step 405, a user simulation script 205 is provided and supplied to the simulation environment. Next, the voice application is processed in step 410 to generate one or more nominal outputs, which may be prompts, synthesized text to speech, pre-recorded audio, etc. The user simulation script is processed in step 315 to generate a simulated output for the voice application corresponding to the nominal output.

[0028] After determining whether the process requires a user input in response to the nominal output from the voice application in step 420, the simulation tool applies conditional logic to the nominal output in step 422. This is done by incorporating one or more conditional statements in the user simulation script and by setting an internal variable in the script to equal the nominal output. Each conditional statement includes a logical test which compares the nominal output to a pre-determined value using the internal variable, and produces a varying result depending on the outcome of the logical test.

[0029] This produces a dynamic environment where a first simulated input can be generated for the voice application in step 425, if, when applying and resolving the logical test in a first conditional statement in step 422, the nominal output satisfies the first condition. If the nominal output does not satisfy the first condition, or satisfies a second condition different from the first condition, a second simulated input for the voice application can be generated in step 425. Of course, either simulated input is pre-determined and incorporated in the user simulation script. Therefore the first simulated input can correspond to a first pre-determined user input to the voice application, while the second simulated input can corresponds to a second pre-determined user input to

the voice application. At step 430, the process may then choose to continue and proceed back to step 410 after assessing whether additional processing of the voice application is necessary. Or it may terminate if execution of the voice application is complete.

[0030] The present invention thereby allows a developer of a voice application to test the application by simulating the real-time flow of events between a user and a voice application. The simulated inputs and outputs are executed in conjunction with the voice application in real-time to test the application. This greatly aids in developing the voice application.

[0031] The present invention can be realized in hardware, software, or a combination of hardware and software. An implementation of the method and system of the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system, or other apparatus adapted for carrying out the methods described herein, is suited to perform the functions described herein.

[0032] A typical combination of hardware and software could be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein. The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described

herein, and which, when loaded in a computer system is able to carry out these methods.

[0033]   Computer program or application in the present context means any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following a) conversion to another language, code or notation; b) reproduction in a different material form. Significantly, this invention can be embodied in other specific forms without departing from the spirit or essential attributes thereof, and accordingly, reference should be had to the following claims, rather than to the foregoing specification, as indicating the scope of the invention.